Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Cancelled)

2. (Currently Amended) An apparatus comprising:

a plurality of inputs to receive multiple input terms;

a multi-stage series of Boolean function generators coupled with the inputs to implement one or more full-adders, half-adders, and single registers to produce intermediate summation results by combining the input terms according to a pipelined reduction pattern~~, the series of Boolean function generators to be structured~~ such that input bits of equal significance are maximally partitioned into groups of three to serve as inputs to full-adders, remaining groups of two to serve as inputs to half-adders, and remaining single bits to serve as inputs to single registers ~~based, at least in part, on one or more attributes of the input terms~~; and

a multi-input adder, logically coupled with the series of Boolean function generators to produce a final sum of the input terms by combining the intermediate summation results.

3. (Previously Presented) The apparatus of claim 2 wherein the input terms include one or more accumulator bits.

4. (Previously Presented) The apparatus of claim 2 wherein the Boolean function generators comprise four-input look-up tables (LUTs) to implement Boolean logic functions.

5. (Currently Amended) The apparatus of claim 2 wherein ~~the~~ each Boolean function generator[[s]] ~~have~~ pairs with an associated register[[s]] to form the atomic structure of a dedicated logic device.

6. (Canceled)

7. (Previously Presented) The apparatus of claim 2 wherein the multi-input adder comprises an adder with an input for each single register in a final stage of the multiple stages of the series.

8. (Previously Presented) The apparatus of claim 7 further comprising:

single registers in the series of Boolean function generators to receive feedback accumulator bits from the multi-input adder, the accumulator bits resulting from a multiply-accumulate operation.

9. (Previously Presented) The apparatus of claim 7 further comprising:

single registers in the series of Boolean function generators to receive feedback accumulator bits resulting from a multiply-accumulate operation; and

an accumulator coupled with the multi-input adder to feed the accumulator bits back into the series of Boolean function generators.

10. (Previously Presented) The apparatus of claim 2 wherein the series of Boolean function generators is incorporated in a dedicated logic device.

11. (Previously Presented) The apparatus of claim 10 wherein the dedicated logic device comprises a field programmable gate array (FPGA).

12. (Previously Presented) The apparatus of claim 10 wherein the dedicated logic device comprises a device with control logic and a block of dedicated logic.

13. (Previously Presented) The apparatus of claim 10 further comprising:

inputs to couple the dedicated logic device with a controller to structure atomic elements of the dedicated logic device into an architecture to implement the one or more full-adders, half-adders, and single registers, the architecture based, at least in part, on an analysis of the input terms.

14. (Previously Presented) The apparatus of claim 10 further comprising:

a logic control module to structure atomic elements of the dedicated logic device into an architecture to implement the one or more full-adders, half-adders, and single registers, the architecture based, at least in part, on an analysis of the input terms.

15. (Previously Presented) The apparatus of claim 14 wherein the analysis of the input terms comprises a bit-wise analysis.

16. (Previously Presented) The apparatus of claim 14 wherein the logic control module dynamically structures the atomic elements of the dedicated logic device to implement desired instances of the architectural structure of the one or more full-adders, half-adders, and single registers.

17. (Currently Amended) A method for performing complex arithmetic, comprising:

receiving a plurality of input terms;

forming a summation pattern by which to reduce the terms by maximally partitioning bits of equal significance into groups of three, remaining bits of equal significance into groups of two, and remaining bits left singly;

~~analyzing an attribute of the input terms;~~

producing intermediate summation results by combining the input terms with a pipelined, multi-stage series of Boolean function generators that implements one or more

full-adders, half-adders, and single registers, the structure of the pipelined series of Boolean function generators based, at least in part, on the groups of bits ~~attribute of the input terms~~; and

producing a final sum of the input terms by combining the intermediate summation results with a multi-input adder.

18. (Previously Presented) The method of claim 17 wherein receiving the plurality of input terms includes receiving one or more accumulator bits.

19. (Previously Presented) The method of claim 17 wherein producing the intermediate summation results by combining the input terms with the multi-stage series of Boolean function generators comprises combining the input terms with four-input look-up tables (LUTs) that implement Boolean logic functions.

20. (Currently Amended) The method of claim 17 wherein ~~producing the intermediate summation results by combining the input terms with the multi-stage series of Boolean function generators comprises combining the input terms~~ each Boolean function generator[[s]] ~~that have~~ pairs with an associated register[[s]] to form the atomic structure of a dedicated logic device.

21. (Currently Amended) The method of claim 17 wherein producing the intermediate summation results ~~by combining the input terms with a multi-stage series of Boolean function generators whose structure is based, at least in part, on the attribute of the input terms~~ comprises combining the input terms with a multi-stage series of Boolean function generators structured to receive three-bit input terms by a full-adder, two-bit input terms by a half-adder, and single-bit input terms by a single register.

22. (Previously Presented) The method of claim 17 wherein producing the final sum with the multi-input adder comprises producing the final sum with an adder that has an input for each single register in a final stage of the multiple stages of the series.

23. (Previously Presented) The method of claim 22 further comprising:

receiving, from the multi-input adder, feedback accumulator bits resulting from a multiply-accumulate operation, with single registers in the series of Boolean function generators.

24. (Previously Presented) The method of claim 22 further comprising:

receiving, from an accumulator coupled with the multi-input adder, feedback accumulator bits resulting from a multiply-accumulate operation, with single registers in the series of Boolean function generators.

25. (Previously Presented) The method of claim 17 wherein producing intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators comprises producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a dedicated logic device.

26. (Previously Presented) The method of claim 25 wherein producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a dedicated logic device comprises producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a field programmable gate array (FPGA).

27. (Previously Presented) The method of claim 25 wherein producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a dedicated logic device comprises producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a device with control logic and a block of dedicated logic.

28. (Previously Presented) The method of claim 25 further comprising:

structuring, with a controller coupled with the dedicated logic device, atomic elements of the dedicated logic device into an architecture to implement the one or more full-adders, half-adders, and single registers, the architecture based, at least in part, on an analysis of the input terms.

29. (Previously Presented) The method of claim 25 further comprising:

structuring atomic elements of the dedicated logic device into an architecture to implement the one or more full-adders, half-adders, and single registers, the architecture based, at least in part, on the analysis of the input terms.

30. (Previously Presented) The method of claim 29 wherein structuring the atomic elements of the dedicated logic device comprises dynamically structuring the atomic elements of the dedicated logic device to implement desired instances of the architectural structure of the one or more full-adders, half-adders, and single registers based on the analysis of the input terms.

31. (Previously Presented) The method of claim 17 wherein analyzing an attribute of the input terms comprises performing a bit-wise analysis of the input terms.

32. (Currently Amended) A method for performing complex ~~arithmetic~~ multiplication comprising:

generating a plurality of partial products from two or more input terms;

for a real-component branch, ~~inverting~~ negating certain partial products and simultaneously passing the ~~inverted~~ negated and ~~non-inverted~~ non-negated partial products to a multi-stage series of Boolean function generators that implements one or more full-adders, half-adders, and single registers to produce intermediate summation results by combining the partial products;

for an imaginary-component branch, simultaneously passing the partial products from the two or more input terms to a multi-stage series of Boolean function generators that implements one or more full-adders, half-adders, and single registers to produce intermediate summation results by combining the partial products;

determining the structure of the Boolean function generators based, at least in part, on one or more attributes of the input terms;

receiving in both branches accumulator bits over a feedback path; and

adding the intermediate summation results with the accumulator bits for each branch to produce a final real-component sum and a final imaginary-component sum.

33. (Previously Presented) The method of claim 32 wherein the method is performed on a dedicated logic device.

34. (Previously Presented) The method of claim 33 wherein the method is performed on a field programmable gate array (FPGA).

35. (Previously Presented) The method of claim 33 wherein the method is performed on a device having control logic and a block of dedicated logic.

36. (Previously Presented) The method of claim 32 wherein generating a plurality of partial product terms comprises combining the two or more inputs in a combinatorial stage of a complex multiply accumulator.

37. (Previously Presented) The method of claim 32 further comprising:

analyzing one or more attributes of the input terms.

38. (Previously Presented) The method of claim 37 wherein analyzing the one or more attributes of the input terms comprises performing a bit-wise analysis of the input terms.

39. (Previously Presented) The method of claim 37 wherein passing partial products to a multi-stage series of Boolean function generators that implements one or more full-adders, half-adders, and single registers further comprises structuring an architecture of the multi-stage series of Boolean function generators based, at least in part, on the analysis of the input terms.

40. (Previously Presented) The method of claim 39 wherein structuring the architecture of the multi-stage series of Boolean function generators comprises structuring atomic elements of a dedicated logic device to implement the architecture.

41. (Previously Presented) The method of claim 39 wherein structuring the architecture of the multi-stage series of Boolean function generators comprises dynamically structuring atomic elements of a dedicated logic device to implement instances of desired architectural structures.

42. (New) The method of claim 32 wherein the multi-stage series of Boolean function generators are pipelined.

43. (New) The method of claim 32 wherein the partial products of the two or more input terms are reduced according to a pattern structured by maximally partitioning bits of equal significance into groups of three to be passed as inputs to the full-adders, remaining groups of two to be passed as inputs to the half-adders, and remaining single bits to be passed to single registers.